

ROBOFUN.RO  
**LECȚIA IX**

# **CURS GRATUIT**

**ARDUINO ȘI ROBOTICĂ**

**Senzori atmosferici**

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs  
CC BY-NC-ND



Codul sursa din acest document este licentiat

Public-Domain

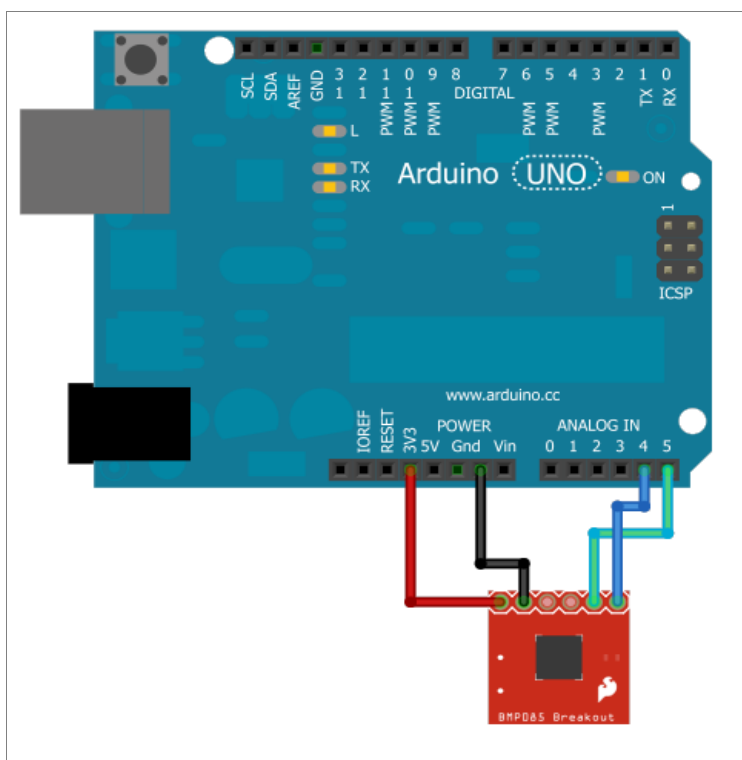
Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

## Senzori Atmosferici

### Presiune Atmosferica – BMP085

Senzorul BMP085 este un senzor foarte precis produs de firma Bosch, capabil sa masoare presiunea atmosferica si temperatura. Cum presiunea atmosferica variaza cu altitudinea, pe baza presiunii atmosferice masurate se poate calcula si altitudinea (lucru foarte util la drone si alte dispozitive zburatoare). Conectarea senzorului la Arduino se face prin I2C, astfel ca avem nevoie de doar doi pini (in afara celor doi pini de alimentare). Pentru a citi valorile senzorului recomand o librarie open-source, disponibila aici : [http://www.robofun.ro/senzor\\_presiune\\_bmp085](http://www.robofun.ro/senzor_presiune_bmp085) (primul link, "Librarie Arduino").

Dupa instalarea librariei, codul sursa este extrem de simplu, ca mai jos.



Arduino 3.3 V	BMP085 VCC
Arduino GND	BMP085 GND
Arduino SDA	BMP085 SDA
Arduino SCL	BMP085 SCL

```
#include <Wire.h>
#include <BMP085.h>

BMP085 dps = BMP085();
long Temperature = 0, Pressure = 0, Altitude = 0;

void setup(void) {
  Serial.begin(9600);
  Wire.begin();
  delay(1000);
  dps.init();
  delay(5000);
}

void loop(void) {
  dps.getTemperature(&Temperature);
  dps.getPressure(&Pressure);
  dps.getAltitude(&Altitude);

  Serial.print("Temp(C):");
  Serial.print(Temperature);
  Serial.print("  Alt(cm):");
  Serial.print(Altitude);
  Serial.print("  Pressure(Pa):");
  Serial.println(Pressure);
}
```

Singurul lucru de remarcat aici este delay-ul de 5 secunde din setup, necesar pentru initializarea senzorului.

## Presiune Atmosferica – MPL115A1

MPL115A1 este destul de similar cu BMP085. Diferă modul de conectare cu Arduino (I2C în cazul BMP085 și SPI în cazul MPL115A1). Conectarea se face folosind pinii SPI, ca mai jos.



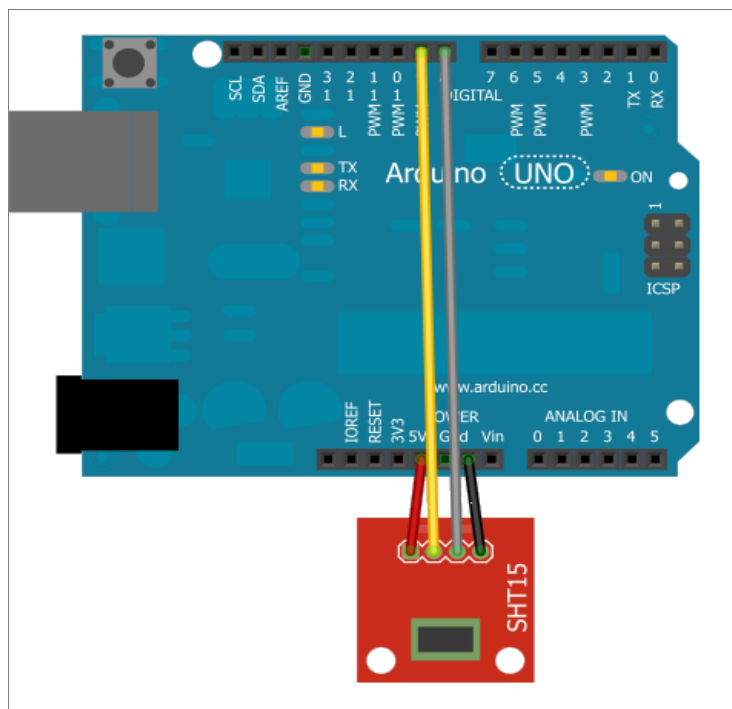
Arduino 3.3 V	MPL115A1 VCC
Arduino GND	MPL115A1 GND
Arduino Digital 9	MPL115A1 SDN
Arduino Digital 10	MPL115A1 CSN
Arduino Digital 12	MPL115A1 SDO
Arduino Digital 11	MPL115A1 SDI
Arduino Digital 13	MPL115A1 SCK

Codul sursa integral în vei găsi la adresa:

[http://www.robofun.ro/senzor\\_presiune\\_atmosferica\\_MPL115A1](http://www.robofun.ro/senzor_presiune_atmosferica_MPL115A1)

## Temperatura si Umiditate – SHT15

SHT15 este un senzor care masoara presiunea si umiditatea mediului ambiant extrem de precis (precizie de 0.3 C, si 2 % RH ). Conectarea la Arduino se face folosind doi pini digitali.



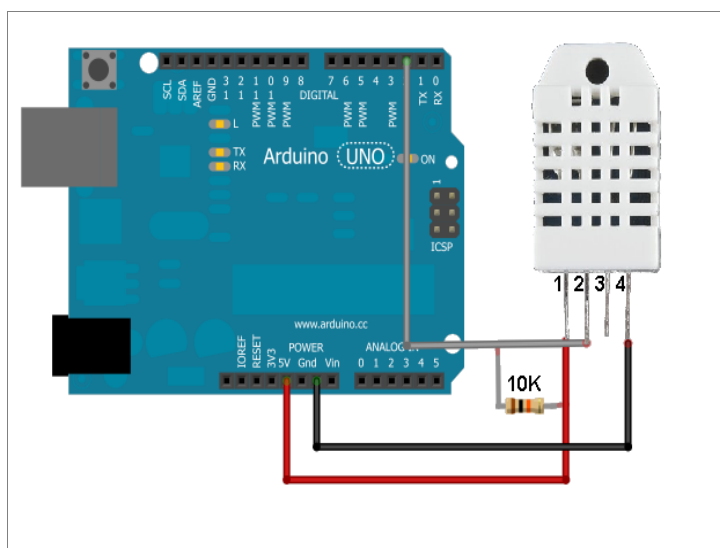
Arduino 5 V	SHT15 VCC
Arduino GND	SHT15 GND
Arduino Digital 9	SHT15 DATA
Arduino Digital 8	SHT15 SCK

Codul sursa integral este disponibil aici - [http://www.robofun.ro/senzor\\_temperatura\\_umiditate\\_sht15](http://www.robofun.ro/senzor_temperatura_umiditate_sht15) .



## Temperatura si Umiditate – DHT22

Daca nu ai nevoie de o precizie chiar asa de mare ca in cazul SHT15 si te multumesti si cu +/- 0.5 C, atunci DHT22 s-ar putea sa fie alegerea corecta pentru tine. Frecventa de citire pentru acest senzor este de asemenea destul de scazuta comparata cu SHT15 (poti face cel mult o citire la 2 secunde). Ca necesar de pini, este foarte modest, avand nevoie doar de un singur pin digital pentru conectarea cu Arduino.



Arduino 5 V	DHT22 VCC
Arduino GND	DHT22 GND
Arduino Digital 2	DHT22 PIN2
Rezistor 10K	conectat intre DHT22 VCC si DHT22 PIN2

Mai departe, instaleaza libraria pentru Arduino disponibila aici :

[http://www.robofun.ro/senzor\\_umiditate\\_temperatura\\_dht22](http://www.robofun.ro/senzor_umiditate_temperatura_dht22)

```
#include <DHT22.h>
#define DHT22_PIN 2

DHT22 myDHT22(DHT22_PIN);

void setup(void)
```

```
{
  Serial.begin(9600);
  Serial.println("DHT22 Library Demo");
}

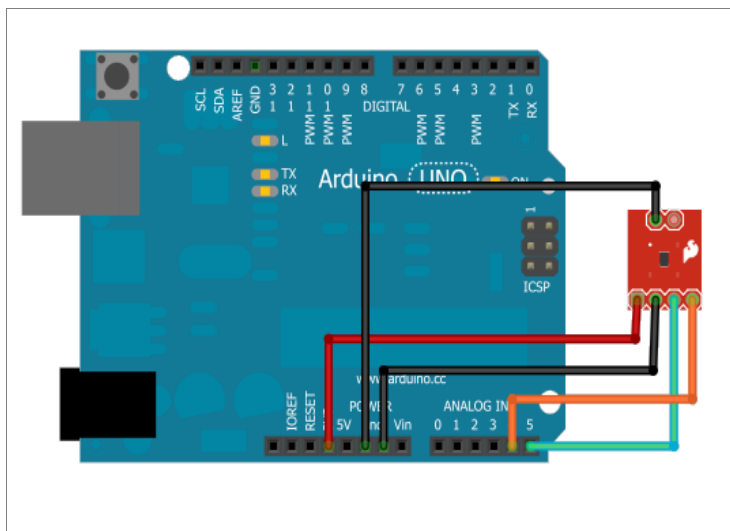
void loop(void) {
  DHT22_ERROR_t errorCode;

  delay(2000);
  Serial.print("Requesting data...");
  errorCode = myDHT22.readData();
  switch(errorCode)
  {
    case DHT_ERROR_NONE:
      Serial.print("Got Data ");
      Serial.print(myDHT22.getTemperatureC());
      Serial.print("C ");
      Serial.print(myDHT22.getHumidity());
      Serial.println("%");
      break;
    case DHT_ERROR_CHECKSUM:
      Serial.print("check sum error ");
      Serial.print(myDHT22.getTemperatureC());
      Serial.print("C ");
      Serial.print(myDHT22.getHumidity());
      Serial.println("%");
      break;
    case DHT_BUS_HUNG:
      Serial.println("BUS Hung ");
      break;
    case DHT_ERROR_NOT_PRESENT:
      Serial.println("Not Present ");
      break;
    case DHT_ERROR_ACK_TOO_LONG:
      Serial.println("ACK time out ");
      break;
    case DHT_ERROR_SYNC_TIMEOUT:
      Serial.println("Sync Timeout ");
      break;
    case DHT_ERROR_DATA_TIMEOUT:
      Serial.println("Data Timeout ");
      break;
    case DHT_ERROR_TOOQUICK:
      Serial.println("Polled to quick ");
      break;
  }
}
```



## Temperatura – TMP102

TMP102 este un senzor de temperatura care comunica pe I2C, cu o rezolutie de 0.0625 C si o precizie de 0.5 C. Alimentarea se face la 3.3 V, iar conectarea la Arduino se face folosind pinii I2C. Limitele sale sunt -25 C pana la 85 C, iar consumul este de-a dreptul impresionant – 10 microAmperi !.



Arduino 3.3 V	TMP102 V+
Arduino GND	TMP102 GND
Arduino SDA (Analog 4)	TMP102 SDA
Arduino SCL (Analog 5)	TMP102 SCL
Arduino GND	TMP102 ADD0

```
#include <Wire.h>
int tmp102Address = 0x48;

void setup(){
  Serial.begin(9600);
  Wire.begin();
}

void loop(){
  float celsius = getTemperature();
  Serial.print("Celsius: ");
  Serial.println(celsius);

  float fahrenheit = (1.8 * celsius) + 32;
  Serial.print("Fahrenheit: ");
```

```
    Serial.println(fahrenheit);

    delay(200);
}

float getTemperature() {
    Wire.requestFrom(tmp102Address, 2);

    byte MSB = Wire.read();
    byte LSB = Wire.read();

    int TemperatureSum = ((MSB << 8) | LSB) >> 4;

    float celsius = TemperatureSum*0.0625;
    return celsius;
}
```

Senzorul TMP102 are 4 adrese distincte posibile, adresa curenta fiind selectata prin conectarea pinului ADD0 la GND, 3.3 V, SDA sau respectiv SCL. Astfel, in exemplul de mai sus, pinul ADD0 este conectat la GND si atunci adresa senzorului este 0x48. Daca vom conecta pinul ADD0 la 3.3 V, atunci adresa senzorului va deveni 0x49. Conectarea pinului ADD0 la pinul SDA va determina adresa 0x4A, iar conectarea la SCL va determina adresa 0x4B. Acest lucru este excelent pentru situatiile in care avem nevoie de mai multi senzori conectati la acelasi Arduino. Putem astfel conecta pana la cel mult patru senzori, toti conectati pe acelasi bus I2C. Pentru primul senzor, pinul ADD0 se conecteaza la GND, pentru al doilea senzor, pinul ADD0 se conecteaza la 3.3V, pentru a treilea senzor, pinul ADD0 se conecteaza la SDA, iar pentru a al patrulea senzor, pinul ADD0 se conecteaza la pinul SCL. Toti ceilalti pini se conecteaza exact ca mai sus.

In cod, vom avea patru adrese distincte, cate o adresa pentru fiecare senzor, ca mai jos.

```
#include <Wire.h>
int tmp102Address1 = 0x48;
int tmp102Address2 = 0x49;
int tmp102Address3 = 0x4A;
int tmp102Address4 = 0x4B;

void setup() {
    Serial.begin(9600);
    Wire.begin();
}

void loop() {
    float celsius = getTemperature(tmp102Address1);
    Serial.print("Celsius, Senzor 1: ");
    Serial.println(celsius);

    celsius = getTemperature(tmp102Address2);
    Serial.print("Celsius, Senzor 2: ");
    Serial.println(celsius);

    celsius = getTemperature(tmp102Address3);
    Serial.print("Celsius, Senzor 3: ");
    Serial.println(celsius);

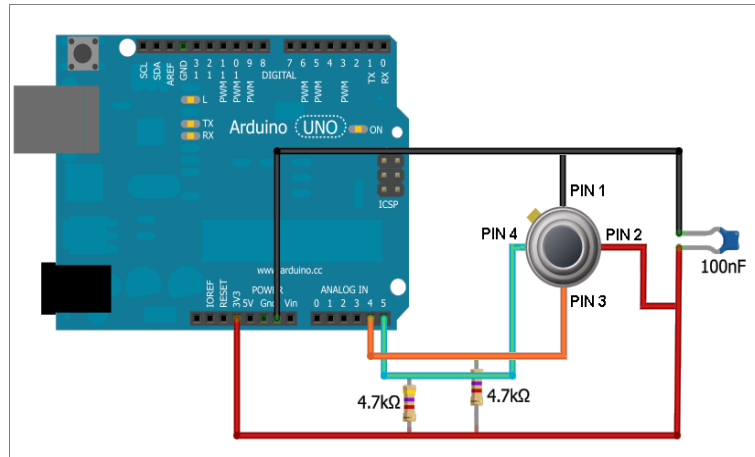
    celsius = getTemperature(tmp102Address4);
```

```
Serial.print("Celsius, Senzor 4: ");  
Serial.println(celsius);  
  
delay(200);  
}  
  
float getTemperature(int address){  
    Wire.requestFrom(address,2);  
  
    byte MSB = Wire.read();  
    byte LSB = Wire.read();  
  
    int TemperatureSum = ((MSB << 8) | LSB) >> 4;  
  
    float celsius = TemperatureSum*0.0625;  
    return celsius;  
}
```

## Temperatura la Distanta – MLX90614

MLX90614 este un senzor de temperatura cu o functionalitate unica. Determina temperatura obiectelor de la distanta, fara a le atinge ! Senzorul functioneaza pe baza determinarii lungimii de unda a radiatiei infrarosie emisa de obiectul in cauza. Determina temperaturi intre -70 C si 382 C, cu o rezolutie imensa (0.0034 C !).

Conectarea la Arduino este ceva mai complexa decat de obicei, pentru ca senzorul nu este inclus pe un montaj, ci montajul va trebui sa il faci tu. Doua rezistoare de 4.7 K si un condensator de 0.1 uF sunt tot ce iti trebuie.



Arduino 3.3 V	MLX90614 PIN2
Arduino GND	MLX90614 PIN1
Arduino SDA (Analog 4)	MLX90614 PIN3
Arduino SCL (Analog 5)	MLX90614 PIN4
Condensator 100nF	Conectat intre MLX90614 PIN1 si MLX90614 PIN2
Rezistor 4.7 K	Conectat intre MLX90614 PIN3 si Arduino 3.3V
Rezistor 4.7 K	Conectat intre MLX90614 PIN4 si Arduino 3.3V

Mai departe, vei avea nevoie sa iti instalezi in mediul tau de dezvoltare Arduino biblioteca I2C Master (disponibila ca download de pe pagina :

[http://www.robofun.ro/senzor\\_infrarosu\\_MLX90614](http://www.robofun.ro/senzor_infrarosu_MLX90614) ).

Codul sursa integral este disponibil mai jos.

```
#include <i2cmaster.h>
void setup(){
  Serial.begin(9600);
  i2c_init();
}
```

```
    PORTC = (1 << PORTC4) | (1 << PORTC5);
}

void loop(){
    int dev = 0x5A<<1;
    int data_low = 0;
    int data_high = 0;
    int pec = 0;

    i2c_start_wait(dev+I2C_WRITE);
    i2c_write(0x07);

    i2c_rep_start(dev+I2C_READ);
    data_low = i2c_readAck();
    data_high = i2c_readAck();
    pec = i2c_readNak();
    i2c_stop();

    double tempFactor = 0.02;
    double tempData = 0x0000;
    int frac;

    tempData = (double)((((data_high & 0x007F) << 8) + data_low));
    tempData = (tempData * tempFactor)-0.01;

    float celsius = tempData - 273.15;
    float fahrenheit = (celsius*1.8) + 32;

    Serial.print("Celsius: ");
    Serial.println(celsius);

    Serial.print("Fahrenheit: ");
    Serial.println(fahrenheit);

    delay(1000);
}
```

---

Aceasta a fost lectia 9. In final, as vrea sa te rog sa ne oferi feedback asupra acestei lectii, pentru a ne permite sa le facem mai bune pe urmatoarele.

Este vorba despre un sondaj cu 4 intrebari (oricare este optionala), pe care il poti accesa [dand click aici](#).

Sau ne poti contacta direct prin email la [contact@robofun.ro](mailto:contact@robofun.ro) .

Iti multumim,

Echipa [Robofun.RO](#)